UNITED STATES PATENT AND TRADEMARK OFFICE

UNITED STATES DEPARTMENT OF COMMERCE
United States Patent and Trademark Office
Address: COMMISSIONER FOR PATENTS
P.O. Box 1450
Alexandria, Virginia 22313-1450
www.uspto.gov

APPLICATION NO.	FILING DATE	FIRST NAMED INVENTOR	ATTORNEY DOCKET NO.	CONFIRMATION NO.
10/730,203	12/04/2003	Wajdi K. Feghali	42P14932	2146

59796 7590 01/29/2008

INTEL CORPORATION
c/o INTELLEVATE, LLC
P.O. BOX 52050
MINNEAPOLIS, MN 55402

EXAMINER	
LOUIE, OSCAR A	

ART UNIT	PAPER NUMBER
2136	

MAIL DATE	DELIVERY MODE
01/29/2008	PAPER

Please find below and/or attached an Office communication concerning this application or proceeding.

The time period for reply, if any, is set in the attached communication.

Office Action Summary	Application No.	Applicant(s)	
	10/730,203	FEGHALI, WAJDI K.	
	Examiner	Art Unit	
	Oscar A. Louie	2136	

-- The MAILING DATE of this communication appears on the cover sheet with the correspondence address --

Period for Reply

A SHORTENED STATUTORY PERIOD FOR REPLY IS SET TO EXPIRE 3 MONTH(S) OR THIRTY (30) DAYS, WHICHEVER IS LONGER, FROM THE MAILING DATE OF THIS COMMUNICATION.

- Extensions of time may be available under the provisions of 37 CFR 1.136(a). In no event, however, may a reply be timely filed after SIX (6) MONTHS from the mailing date of this communication.
- If NO period for reply is specified above, the maximum statutory period will apply and will expire SIX (6) MONTHS from the mailing date of this communication.
- Failure to reply within the set or extended period for reply will, by statute, cause the application to become ABANDONED (35 U.S.C. § 133). Any reply received by the Office later than three months after the mailing date of this communication, even if timely filed, may reduce any earned patent term adjustment. See 37 CFR 1.704(b).

Status

1)☒ Responsive to communication(s) filed on *19 November 2007*.

2a)☒ This action is **FINAL**. 2b)☐ This action is non-final.

3)☐ Since this application is in condition for allowance except for formal matters, prosecution as to the merits is closed in accordance with the practice under *Ex parte Quayle*, 1935 C.D. 11, 453 O.G. 213.

Disposition of Claims

4)☒ Claim(s) *1-41* is/are pending in the application.

4a) Of the above claim(s) _____ is/are withdrawn from consideration.

5)☐ Claim(s) _____ is/are allowed.

6)☒ Claim(s) *1-41* is/are rejected.

7)☐ Claim(s) _____ is/are objected to.

8)☐ Claim(s) _____ are subject to restriction and/or election requirement.

Application Papers

9)☐ The specification is objected to by the Examiner.

10)☐ The drawing(s) filed on _____ is/ are: a)☐ accepted or b)☐ objected to by the Examiner.

Applicant may not request that any objection to the drawing(s) be held in abeyance. See 37 CFR 1.85(a).

Replacement drawing sheet(s) including the correction is required if the drawing(s) is objected to. See 37 CFR 1.121(d).

11)☐ The oath or declaration is objected to by the Examiner. Note the attached Office Action or form PTO-152.

Priority under 35 U.S.C. § 119

12)☐ Acknowledgment is made of a claim for foreign priority under 35 U.S.C. § 119(a)-(d) or (f).

a)☐ All b)☐ Some * c)☐ None of:

1.☐ Certified copies of the priority documents have been received.

2.☐ Certified copies of the priority documents have been received in Application No. _____.

3.☐ Copies of the certified copies of the priority documents have been received in this National Stage application from the International Bureau (PCT Rule 17.2(a)).

* See the attached detailed Office action for a list of the certified copies not received.

Attachment(s)

1) ☐ Notice of References Cited (PTO-892)
2) ☐ Notice of Draftsperson's Patent Drawing Review (PTO-948)
3) ☐ Information Disclosure Statement(s) (PTO/SB/08) Paper No(s)/Mail Date _____.
4) ☐ Interview Summary (PTO-413) Paper No(s)/Mail Date. _____.
5) ☐ Notice of Informal Patent Application
6) ☐ Other: _____.

DETAILED ACTION

This final action is in response to the amendment filed on 11/19/2007. Claims 1-41 are pending and have been considered as follows.

Examiner's Note

1. The Applicant appears to be attempting to invoke 35 U.S.C. 112 6th paragraph in Claim 35 by using "means-plus-function" language. However, the Examiner notes that the only "means" for performing these cited functions in the specification appears to be computer program modules. While the claims pass the first test of the three-prong test used to determine invocation of paragraph 6, since no other specific structural limitations are disclosed in the specification, the claims do not meet the other tests of the three-prong test. Therefore, 35 U.S.C. 112 6th paragraph has not been invoked when considering these claims below.

Claim Rejections - 35 USC § 102

2. The following is a quotation of the appropriate paragraphs of 35 U.S.C. 102 that form the basis for the rejections under this section made in this Office action:

> A person shall be entitled to a patent unless –
>
> (e) the invention was described in (1) an application for patent, published under section 122(b), by another filed in the United States before the invention by the applicant for patent or (2) a patent granted on an application for patent by another filed in the United States before the invention by the applicant for patent, except that an international application filed under the treaty defined in section 351(a) shall have the effects for purposes of this subsection of an application filed in the United States only if the international application designated the United States and was published under Article 21(2) of such treaty in the English language.

3. Claims 1-8, 15-17, 21-24, & 27-34 are rejected under 35 U.S.C. 102(e) as being anticipated by Buer et al (US-7177421-B2)

Claim 1:

Buer et al disclose a processor comprising,

- "a plurality of pipeline stages to perform an inner loop of a hash algorithm" (i.e. "The engine architecture implements a pipelined structure to hide the time required for performing the outer hash operation when multiple data payloads are fed to the engine continuously. The engine architecture includes a core having two instantiations of the hash round logic; in this instance, inner and outer hash engines (inner and outer loops) for each of the MD5 hash round logic and the SHA1 hash round logic supported by the IPSec protocol") [column 6 lines 18-25];
- "the plurality of pipeline stages comprising at least as many pipeline stages as there are iterations of the inner loop to be performed" (i.e. "Pipeline control logic ensures that the outer hash operation for one data payload is performed in parallel with the inner hash

operation of the next data payload in the packet stream fed to the authentication engine. A dual-frame input buffer is used for the inner hash engine, allowing one new 512-bit block to be loaded while another one is being processed, and the initial hash states are double buffered for concurrent inner hash and outer hash operations") [column 6 lines 25-33].

Claim 2:

Buer et al disclose a processor, as in Claim 1 above, further comprising,

- "the plurality of pipeline stages further comprises as many pipeline stages as there are chaining variables to be used in the inner loop" (i.e. "The engine architecture includes a core having two instantiations of the hash round logic; in this instance, inner and outer hash engines (inner and outer loops) for each of the MD5 hash round logic and the SHA1 hash round logic supported by the IPSec protocol") [column 6 lines 18-25].

Claim 3:

Buer et al disclose a processor, as in Claim 2 above, further comprising,

- "each pipeline stage comprises an adder, shifter, and logic to perform a function" (i.e. "For each 512-bit data block, a set of operations including non-linear functions, shift functions and additions, called a "round," is applied to the block repeatedly") [column 2 lines 27-29].

Claim 4:

Buer et al disclose a processor, as in Claim 3 above, further comprising,

- "control logic to schedule operations to be executed within the plurality of pipeline stages" (i.e. "The architecture is configured to pipeline the hash operations of the inner hash and outer hash engines, collapse and rearrange multi-round logic to reduce rounds of hash operations, and implement multi-round logic to schedule addition computations to be conducted in parallel with round operations") [column 4 lines 29-34].

Claim 5:

Buer et al disclose a processor, as in Claim 4 above, further comprising,

- "operations are to be scheduled by the control logic and executed by the plurality of pipeline stages so as to minimize data dependencies between iterations of the inner loop to be performed" (i.e. "The architecture is configured to pipeline the hash operations of the inner hash and outer hash engines, collapse and rearrange multi-round logic to reduce rounds of hash operations, and implement multi-round logic to schedule addition computations to be conducted in parallel with round operations") [column 4 lines 29-34].

Claim 6:

Buer et al disclose a processor, as in Claim 5 above, further comprising,

- "the hash algorithm is chosen from a group of secure hash algorithms (SHA) consisting of SHA-1, SHA-128, SHA-196. SHA-256, and message digest 5 (MD5)" (i.e. "Both MD5 and SHA1 specify that data is to be processed in 512-bit blocks") [column 2 lines 4-5].

Claim 7:

Buer et al disclose a processor, as in Claim 6 above, further comprising,

- "the hash algorithm is to be performed at an operating frequency equal to that of the adder" (i.e. "Pipeline control logic ensures that the outer hash operation for one data payload is performed in parallel with the inner hash operation of the next data payload in the packet stream fed to the authentication engine. A dual-frame input buffer is used for the inner hash engine, allowing one new 512-bit block to be loaded while another one is being processed, and the initial hash states are double buffered for concurrent inner hash and outer hash operations. In addition, dual-ported ROM is used for concurrent constant lookups by both inner and outer hash engines") [column 6 lines 25-35].

Claim 8:

Buer et al disclose a processor, as in Claim 7 above, further comprising,

- "the plurality of pipeline stages comprises 88 pipeline stages to process 512 bits of data" (i.e. "In a preferred embodiment, the eighty rounds of an SHA1 loop are collapsed into forty rounds. As described and illustrated above, the collapsing of rounds is accomplished by having a single set of registers (the preferred embodiment has 5 registers as defined by the IPSec protocol) with two rounds of logic. It is contemplated that the techniques of invention described herein can also be applied to further collapse the number of SHA1 rounds in an SHA1 loop into twenty or even fewer rounds") [column 8 lines 13-21].

Claim 15:

Buer et al disclose a machine-readable medium having stored thereon a set of instructions, which if executed by a machine cause the machine to perform a method comprising,

- "performing a plurality of iterations of an inner loop of an hash algorithm in parallel, the plurality of iterations performed in parallel being limited, at least in part, by dependencies between each of the plurality of iterations of the inner loop" (i.e. "The engine architecture implements a pipelined structure to hide the time required for performing the outer hash operation when multiple data payloads are fed to the engine continuously. The engine architecture includes a core having two instantiations of the hash round logic; in this instance, inner and outer hash engines (inner and outer loops) for each of the MD5 hash round logic and the SHA1 hash round logic supported by the IPSec protocol") [column 6 lines 18-25];
- "adding initial values of a plurality of chaining variables to final values of the plurality of chaining variables, the final values being a result of performing the plurality of iterations of the inner loop" (i.e. "As noted above with reference to FIGS. 3 and 4, in both MD5 and SHA1, only one state register is re-computed every round. The rest of the state registers use shifted or non-shifted contents from neighboring registers. Thus, the final hash states are not generated in the final round, but rather in the last four consecutive MD5 rounds or five SHA1 rounds, respectively. The present invention exploits this observation by providing architecture and logic enabling the scheduling of the additions as early as the final hash state is available, hiding the computation time completely behind the round operations") [column 8 lines 34-44].

Claim 16:

Buer et al disclose a machine-readable medium having stored thereon a set of instructions, which if executed by a machine cause the machine to perform a method, as in Claim 15 above, further comprising,

- "controlling scheduling of operations performed as a result of performing the plurality of iterations of the inner loop. the scheduling being controlled so as to minimize a critical path among the operation" (i.e. "The present invention exploits this observation by providing architecture and logic enabling the scheduling of the additions as early as the final hash state is available, hiding the computation time completely behind the round operations") [column 8 lines 40-44].

Claim 17:

Buer et al disclose a machine-readable medium having stored thereon a set of instructions, which if executed by a machine cause the machine to perform a method, as in Claim 16 above, further comprising,

- "the critical path depends upon the dependencies between the plurality of iterations of the inner loop" (i.e. "In accordance with this invention, in successive SHA1 rounds the registers having the critical path are alternative so that four registers worth of data may always be passed on to the next round prior to completion of the critical path in the current round") [column 7 lines 65-67 & column 2 lines 1-2].

Claim 21:

Buer et al disclose a method comprising,

- "performing a hash algorithm within a pipelined processor by performing a plurality of iterations of an inner loop of the hash algorithm in parallel" (i.e. "The engine architecture implements a pipelined structure to hide the time required for performing the outer hash operation when multiple data payloads are fed to the engine continuously. The engine architecture includes a core having two instantiations of the hash round logic; in this instance, inner and outer hash engines (inner and outer loops) for each of the MD5 hash round logic and the SHA1 hash round logic supported by the IPSec protocol") [column 6 lines 18-25];
- "generating a plurality of output data elements as a result of performing the hash algorithm" [Fig 2 discloses the generation of output of hash results].

Claim 22:

Buer et al disclose a method, as in claim 21 above, further comprising,

- "scheduling operations associated with the plurality of iterations so as to facilitate a maximum number of the operations to be performed in parallel" (i.e. "The architecture is configured to pipeline the hash operations of the inner hash and outer hash engines, collapse and rearrange multi-round logic to reduce rounds of hash operations, and implement multi-round logic to schedule addition computations to be conducted in parallel with round operations") [column 4 lines 29-34].

Claim 23:

Buer et al disclose a method, as in claim 22 above, further comprising,

- "the maximum number depends upon dependencies between the operations" (i.e. "In every round, the operation starts with certain hash states (referred to as "context") held by hash state registers (in hardware) or variables (in software), and ends with a new set of hash states (i.e., an initial "set" of hash states and an end set; a "set" may be of 4 or 5 for the number of registers used by MD5 and SHA1, respectively). MD5 and SHA1 each specify a set of constants as the initial hash states for the first 512-bit block. The following blocks use initial hash states resulting from additions of the initial hash states and the ending hash states of the previous blocks") [column 2 lines 32-42].

Claim 24:

Buer et al disclose a method, as in claim 22 above, further comprising,

- "wherein the output data elements are transmitted within a computer network" (i.e. "In general, the present invention provides an architecture (hardware implementation) for an authentication engine to increase the speed at which multi-loop and/or multi-round authentication algorithms may be performed on data packets transmitted over a computer network") [column 5 lines 38-42].

Claim 27:

Buer et al disclose a system comprising,

- "a memory unit to store operations of a hash algorithm" [Fig 2 discloses a memory];

- "a pipelined processor to perform the operations of the hash algorithm by performing iterations of an inner loop of the hash algorithm within separate pipeline stages of the pipelined processor" (i.e. "receiving a data packet stream, splitting the packet data stream into fixed-size data blocks, and processing the fixed-size data blocks using a multi-loop, multi-round authentication engine architecture having a hash engine core with an inner hash engine and an outer hash engine. The architecture is configured to pipeline the hash operations of the inner hash and outer hash engines, collapse and rearrange multi-round logic to reduce rounds of hash operations, and implement multi-round logic to schedule addition computations to be conducted in parallel with round operation") [column 4 lines 22-34].

Claim 28:

Buer et al disclose a system, as in Claim 27 above, further comprising,

- "the operations are scheduled so as to minimize the number dependencies among the operation" (i.e. "The architecture is configured to pipeline the hash operations of the inner hash and outer hash engines, collapse and rearrange multi-round logic to reduce rounds of hash operations, and implement multi-round logic to schedule addition computations to be conducted in parallel with round operations") [column 4 lines 29-34].

Claim 29:

Buer et al disclose a system, as in Claim 28 above, further comprising,

- "a bus upon which to drive data generated by performing the hash algorithm within the pipelined processor" [Fig 2 discloses a bus].

Claim 30:

Buer et al disclose a system, as in Claim 28 above, further comprising,

- "a bus to receive data to be operated on by the pipelined processor to perform the hash algorithm" [Fig 2 discloses a bus].

Claim 31:

Buer et al disclose a system, as in Claim 30 above, further comprising,

- "512 bits of data is to be processed by at least 83 pipeline stages of the pipelined processor" (i.e. "In a preferred embodiment, the eighty rounds of an SHA1 loop are collapsed into forty rounds. As described and illustrated above, the collapsing of rounds is accomplished by having a single set of registers (the preferred embodiment has 5 registers as defined by the IPSec protocol) with two rounds of logic. It is contemplated that the techniques of invention described herein can also be applied to further collapse the number of SHA1 rounds in an SHA1 loop into twenty or even fewer rounds") [column 8 lines 13-21].

Claim 32:

Buer et al disclose a system, as in Claim 27 above, further comprising,

- "the pipelined processor is a network processor coupled to a network" (i.e. "In general, the present invention provides an architecture (hardware implementation) for an authentication engine to increase the speed at which multi-loop and/or multi-round authentication algorithms may be performed on data packets transmitted over a computer network") [column 5 lines 38-42].

Claim 33:

Buer et al disclose a system, as in Claim 32 above, further comprising,

- "a host processor coupled to the network processor to perform a portion of the hash algorithm" (i.e. "In general, the present invention provides an architecture (hardware implementation) for an authentication engine to increase the speed at which multi-loop and/or multi-round authentication algorithms may be performed on data packets transmitted over a computer network") [column 5 lines 38-42].

Claim 34:

Buer et al disclose a system, as in Claim 27 above, further comprising,

"the hash algorithm is chosen from a group of secure hash algorithms (SHA) consisting of SHA-1, SHA-128, SHA-196, SHA-256, and message digest 5 (MD5)" (i.e. "Both MD5 and SHA1 specify that data is to be processed in 512-bit blocks") [column 2 lines 4-5].

Claim Rejections - 35 USC § 103

4. The following is a quotation of 35 U.S.C. 103(a) which forms the basis for all obviousness rejections set forth in this Office action:

> (a) A patent may not be obtained though the invention is not identically disclosed or described as set forth in section 102 of this title, if the differences between the subject matter sought to be patented and the prior art are such that the subject matter as a whole would have been obvious at the time the invention was made to a person having ordinary skill in the art to which said subject matter pertains. Patentability shall not be negatived by the manner in which the invention was made.

5. Claims 9-13 & 35-41 are rejected under 35 U.S.C. 103(a) as being unpatentable over Buer et al (US-7177421-B2) in view of M.J.B. Robshaw ("On Recent Results for MD2, MD4, and MD5").

Claim 9 :

Buer et al disclose an apparatus comprising,

- "a first plurality of pipeline stages to perform a hash" (i.e. "As described above, both MD5 and SHA1 algorithms specify that the final hash states of every 512-bit block are to be added together with the initial hash states. The results are then used as the initial states of the next 512-bit block. In MD5, values of four pairs of 32-bit registers need to be added and in SHA1, five pairs") [column 8 lines 23-28];

but Buer et al do not disclose,

- "a first pipeline stage to add a first constant to a first data word to yield a first result"
- "a second pipeline stage to add the first result a first chaining variable, perform a first function on a second, third, and fourth chaining variable to yield a second result, and add the first constant to a second data word to yield a third result"

- "a third pipeline stage to add the second result to the sum of a fifth chaining variable and the first result, add the first constant to a third data word, add the third result to the fourth chaining variable, perform the first function on the first, second. and third chaining variables after they each of have been shifted by a plurality of bits"
- "a second plurality of pipeline stages to add an initial state of the first, second, third, fourth, and fifth chaining variables to a final state of the first, second, third, fourth, and fifth chaining variables, respectively"

however, M.J.B. Robshaw does disclose,

- "Most hash functions have a similar iterative structure which is based around what is termed a compression function [4, 14]. In short, the computation of the hash value for some message depends on what is called a chaining variable. At the start of hashing, this chaining variable has a fixed initial value which is specified as part of the algorithm. The compression function is then used to update the value of this chaining variable in a suitably complex way under the action and influence of part of the message being hashed. This process continues recursively, with the chaining variable being updated under the action of different parts of the message, until all the message (and any additional padding specified by the algorithm) has been used. The final value of the chaining variable is then output as the hash value corresponding to that message" [page 2 column 2];

Therefore, it would have been obvious for one of ordinary skill in the art to at the time of the applicant's invention to include, "a first pipeline stage to add a first constant to a first data word to yield a first result" and "a second pipeline stage to add the first result a first chaining variable, perform a first function on a second, third, and fourth chaining variable to yield a second result,

and add the first constant to a second data word to yield a third result" and "a third pipeline stage to add the second result to the sum of a fifth chaining variable and the first result, add the first constant to a third data word, add the third result to the fourth chaining variable, perform the first function on the first, second. and third chaining variables after they each of have been shifted by a plurality of bits" and "a second plurality of pipeline stages to add an initial state of the first, second, third, fourth, and fifth chaining variables to a final state of the first, second, third, fourth, and fifth chaining variables, respectively," in the invention as disclosed by Buer et al since the limitations as disclosed by M.J.B. Robshaw are common procedures for the calculations of hash functions (i.e. MD5, SHA, etc).

Claim 10:

Buer et al and M.J.B. Robshaw disclose an apparatus, as in Claim 9 above, further comprising,

- "the first plurality of pipeline stages comprises 83 pipeline stages to process 512 bits of information" (i.e. "In a preferred embodiment, the eighty rounds of an SHA1 loop are collapsed into forty rounds. As described and illustrated above, the collapsing of rounds is accomplished by having a single set of registers (the preferred embodiment has 5 registers as defined by the IPSec protocol) with two rounds of logic. It is contemplated that the techniques of invention described herein can also be applied to further collapse the number of SHA1 rounds in an SHA1 loop into twenty or even fewer rounds") [column 8 lines 13-21].

Claim 11:

Buer et al and M.J.B. Robshaw disclose an apparatus, as in Claim 9 above, the combination further disclose,

- "the second plurality of pipeline stages comprises 5 pipeline stages to process 512 bits of information" (i.e. "In a preferred embodiment, the eighty rounds of an SHA1 loop are collapsed into forty rounds. As described and illustrated above, the collapsing of rounds is accomplished by having a single set of registers (the preferred embodiment has 5 registers as defined by the IPSec protocol) with two rounds of logic. It is contemplated that the techniques of invention described herein can also be applied to further collapse the number of SHA1 rounds in an SHA1 loop into twenty or even fewer rounds") [column 8 lines 13-21].

Claim 12:

Buer et al and M.J.B. Robshaw disclose an apparatus, as in Claim 9 above, the combination further disclose,

- "the first and second plurality of pipeline stages are implemented within a network processor architecture" (i.e. "Input data payloads received by the engine 200, for example from data packets received off a network by a chip on which the engine architecture is implemented, are distributed between the frames 202, 204 of the input data buffer 201 so that one data block may be loaded into the buffer while another one is being processed downstream in the data flow") [column 6 lines 38-44].

Claim 13:

Buer et al and M.J.B. Robshaw disclose an apparatus, as in Claim 9 above, the combination further disclose,

- "the hash algorithm is a secure hash algorithm (SHA) and the plurality bits is 30" (i.e. "As noted above with reference to FIGS. 3 and 4, in both MD5 and SHA1, only one state register is re-computed every round. The rest of the state registers use shifted or non-shifted contents from neighboring registers. Thus, the final hash states are not generated in the final round, but rather in the last four consecutive MD5 rounds or five SHA1 rounds, respectively") [column 8 lines 34-40].

Claim 35:

Buer et al disclose an apparatus comprising,

- "execution means for performing iterations of an inner loop of a hash algorithm in parallel" (i.e. "The engine architecture implements a pipelined structure to hide the time required for performing the outer hash operation when multiple data payloads are fed to the engine continuously. The engine architecture includes a core having two instantiations of the hash round logic; in this instance, inner and outer hash engines (inner and outer loops) for each of the MD5 hash round logic and the SHA1 hash round logic supported by the IPSec protocol") [column 6 lines 18-25].

- "scheduling means for scheduling operations associated with the hash algorithm" (i.e. "The architecture is configured to pipeline the hash operations of the inner hash and outer hash engines, collapse and rearrange multi-round logic to reduce rounds of hash operations, and implement multi-round logic to schedule addition computations to be conducted in parallel with round operations") [column 4 lines 29-34];

but Buer et al do not disclose,

- "first means for adding a first constant to a first data word to yield a first result"
- "second means for adding the first result a first chaining variable, performing a first function on a second, third, and fourth chaining variable to yield a second result, and adding the first constant to a second data word to yield a third result"
- "third means for adding the second result to the sum of a fifth chaining variable and the first result, adding the first constant to a third data word, adding the third result to the fourth chaining variable, performing the first function on the first, second, and third chaining variables after they each of have been shifted by a plurality of bits"
- "adding means for adding an initial state of the first, second, third, fourth, and fifth chaining variables to a final state of the first, second, third, fourth, and fifth chaining variables, respectively"

however, M.J.B. Robshaw does disclose,

- "Most hash functions have a similar iterative structure which is based around what is termed a compression function [4, 14]. In short, the computation of the hash value for some message depends on what is called a chaining variable. At the start of hashing, this chaining variable has a fixed initial value which is specified as part of the algorithm. The

compression function is then used to update the value of this chaining variable in a suitably complex way under the action and influence of part of the message being hashed. This process continues recursively, with the chaining variable being updated under the action of different parts of the message, until all the message (and any additional padding specified by the algorithm) has been used. The final value of the chaining variable is then output as the hash value corresponding to that message" [page 2 column 2];

Therefore, it would have been obvious for one of ordinary skill in the art to at the time of the applicant's invention to include, "first means for adding a first constant to a first data word to yield a first result" and "second means for adding the first result a first chaining variable, performing a first function on a second, third, and fourth chaining variable to yield a second result, and adding the first constant to a second data word to yield a third result" and "third means for adding the second result to the sum of a fifth chaining variable and the first result, adding the first constant to a third data word, adding the third result to the fourth chaining variable, performing the first function on the first, second, and third chaining variables after they each of have been shifted by a plurality of bits" and "adding means for adding an initial state of the first, second, third, fourth, and fifth chaining variables to a final state of the first, second, third, fourth, and fifth chaining variables, respectively," in the invention as disclosed by Buer et al since the limitations as disclosed by M.J.B. Robshaw are common procedures for the calculations of hash functions (i.e. MD5, SHA, etc).

Claim 36:

Buer et al and M.J.B. Robshaw disclose an apparatus, as in Claim 35 above, the combination further disclose,

- "the execution means is a pipelined architecture and wherein each of the first, second, and third means are pipeline stages of the pipelined architecture" (i.e. "The architecture is configured to pipeline the hash operations of the inner hash and outer hash engines, collapse and rearrange multi-round logic to reduce rounds of hash operations, and implement multi-round logic to schedule addition computations to be conducted in parallel with round operations") [column 4 lines 29-34].

Claim 37:

Buer et al and M.J.B. Robshaw disclose an apparatus, as in Claim 35 above, the combination further disclose,

- "the scheduling means is a controller to schedule operations associated with the inner loop according to dependencies among the operations" (i.e. "The architecture is configured to pipeline the hash operations of the inner hash and outer hash engines, collapse and rearrange multi-round logic to reduce rounds of hash operations, and implement multi-round logic to schedule addition computations to be conducted in parallel with round operations") [column 4 lines 29-34].

Claim 38:

Buer et al and M.J.B. Robshaw disclose an apparatus, as in Claim 36 above, the combination further disclose,

- "each iteration of the inner loop requires three pipeline stages to perform the iteration" (i.e. "The architecture is configured to pipeline the hash operations of the inner hash and outer hash engines, collapse and rearrange multi-round logic to reduce rounds of hash operations, and implement multi-round logic to schedule addition computations to be conducted in parallel with round operations") [column 4 lines 29-34].

Claim 39:

Buer et al and M.J.B. Robshaw disclose an apparatus, as in Claim 38 above, but Buer et al do not disclose,

- "the adding means comprises the same number of pipeline stages as chaining variables"

however, M.J.B. Robshaw does disclose,

- "Most hash functions have a similar iterative structure which is based around what is termed a compression function [4, 14]. In short, the computation of the hash value for some message depends on what is called a chaining variable. At the start of hashing, this chaining variable has a fixed initial value which is specified as part of the algorithm. The compression function is then used to update the value of this chaining variable in a suitably complex way under the action and influence of part of the message being hashed. This process continues recursively, with the chaining variable being updated under the

action of different parts of the message, until all the message (and any additional padding specified by the algorithm) has been used. The final value of the chaining variable is then output as the hash value corresponding to that message" [page 2 column 2];

Therefore, it would have been obvious for one of ordinary skill in the art at the time of the applicant's invention to include, "the adding means comprises the same number of pipeline stages as chaining variables," in the invention as disclosed by Buer et al since the number of pipeline stages is dependent on the chaining variable.

Claim 40:

Buer et al and M.J.B. Robshaw disclose an apparatus, as in Claim 35 above, the combination further disclose,

- "the hash algorithm is chosen from a group of secure hash algorithms (SHA) consisting of SHA-I, SMA-128, SHA-196, SHA-256, and message digest 5 (MD5)" (i.e. "Both MD5 and SHA1 specify that data is to be processed in 512-bit blocks") [column 2 lines 4-5].

Claim 41:

Buer et al and M.J.B. Robshaw disclose an apparatus, as in Claim 35 above, but Buer et al do not disclose,

- "the plurality of bits is 30"

however, M.J.B. Robshaw does disclose,

- "The compression function is then used to update the value of this chaining variable in a suitably complex way under the action and influence of part of the message being hashed" [page 2 column 2];

Therefore, it would have been obvious for one of ordinary skill in the art at the time of the applicant's invention to include, "the plurality of bits is 30," in the invention as disclosed by Buer et al since the plurality of bits is dependent upon the chaining variable.

6. Claims 18-20, 25 & 26 are rejected under 35 U.S.C. 103(a) as being unpatentable over Buer et al (US-7177421-B2) in view of Bianco et al (US-5365588-A).

Claim 18:

Buer et al disclose a machine-readable medium having stored thereon a set of instructions, which if executed by a machine cause the machine to perform a method, as in Claim 17 above, but do not disclose,

- "decoding the inner loop of the hash algorithm into a first number of operational stages, the first number of operational stages being equal to at least the plurality of iteration"

however, Bianco et al do disclose,

- "the number of stages in the working register should be at least half of the number of taps for the output functions, N times T. For example, with 8 output functions and 6 input taps for each function, the working register should have at least 24 stages" [column 6 lines 20-24];

Therefore, it would have been obvious for one of ordinary skill in the art at the time of the applicant's invention to include, "decoding the inner loop of the hash algorithm into a first number of operational stages, the first number of operational stages being equal to at least the plurality of iteration," in the invention as disclosed by Buer et al for the purposes of throughput.

Claim 19:

Buer et al and Bianco et al disclose a machine-readable medium having stored thereon a set of instructions, which if executed by a machine cause the machine to perform a method, as in Claim 18 above, further comprising,

- "the inner loop is to be performed to process a first number of data elements transmitted over a network" (i.e. "In general, the present invention provides an architecture (hardware implementation) for an authentication engine to increase the speed at which multi-loop and/or multi-round authentication algorithms may be performed on data packets transmitted over a computer network") [column 5 lines 38-42].

Claim 20:

Buer et al and Bianco et al disclose a machine-readable medium having stored thereon a set of instructions, which if executed by a machine cause the machine to perform a method, as in Claim 19 above, further comprising,

- "the first number of operational stages is at least 83 and the first number of data elements comprises 512 bits" (i.e. "In a preferred embodiment, the eighty rounds of an SHA1 loop are collapsed into forty rounds. As described and illustrated above, the collapsing of rounds is accomplished by having a single set of registers (the preferred embodiment has 5 registers as defined by the IPSec protocol) with two rounds of logic. It is contemplated that the techniques of invention described herein can also be applied to further collapse the number of SHA1 rounds in an SHA1 loop into twenty or even fewer rounds") [column 8 lines 13-21].

Claim 25:

Buer et al disclose a method, as in Claim 24 above, but do not disclose,

- "wherein the hash algorithm is performed at the operating frequency of the processor"

however, Bianco et al do disclose,

- "For example, to operate continuously for 365 days at one gigahertz without repeating, the number of cells in the working register must be at least 55. Second, key patterns are easier to determine if the number of stages in the working register is equal to the exponents which produce Mersenne primes" [column 6 lines 11-16];

Therefore, it would have been obvious for one of ordinary skill in the art at the time of the applicant's invention to include, "wherein the hash algorithm is performed at the operating frequency of the processor," in the invention as disclosed by Buer et al since the operating frequency is dependent on the desired amount of operations to be performed and the total number of available cells in a working register.

Claim 26:

Buer et al disclose a method, as in Claim 24 above, but do not disclose,

- "the hash algorithm is performed at 1.4GHz"

however, Bianco et al do disclose,

- "For example, to operate continuously for 365 days at one gigahertz without repeating, the number of cells in the working register must be at least 55. Second, key patterns are easier to determine if the number of stages in the working register is equal to the exponents which produce Mersenne primes" [column 6 lines 11-16];

Therefore, it would have been obvious for one of ordinary skill in the art at the time of the applicant's invention to include, "the hash algorithm is performed at 1.4GHz," in the invention as disclosed by Buer et al since the operating frequency is dependent on the desired amount of operations to be performed and the total number of available cells in a working register.

7. Claim 14 is rejected under 35 U.S.C. 103(a) as being unpatentable over Buer et al (US-7177421-B2) in view of M.J.B. Robshaw ("On Recent Results for MD2, MD4, and MD5") in further view of Bianco et al (US-5365588-A).

Claim 14:

Buer et al and M.J.B. Robshaw disclose an apparatus, as in Claim 9 above, but do not disclose,

- "the network processor architecture is to perform the hash algorithm at an operating frequency of at least 1.4 G.Hz"

however, Bianco et al do disclose,

- "For example, to operate continuously for 365 days at one gigahertz without repeating, the number of cells in the working register must be at least 55. Second, key patterns are easier to determine if the number of stages in the working register is equal to the exponents which produce Mersenne primes" [column 6 lines 11-16];

Therefore, it would have been obvious for one of ordinary skill in the art at the time of the applicant's invention to include, "the network processor architecture is to perform the hash algorithm at an operating frequency of at least 1.4 GHz," in the invention as disclosed by the combination of Buer et al and M.J.B. Robshaw since the operating frequency is dependent on the desired amount of operations to be performed and the total number of available cells in a working register.

Response to Arguments

8. Applicant's arguments filed 11/19/2007 have been fully considered but they are not persuasive.

- The applicant's argument that states, "Buer et al. describes pipelining an iteration of the outer loop with an iteration of the inner loop, not pipelining iterations of the inner loop," has been considered but is non-persuasive. The examiner notes that Buer et al. disclose, "inner and outer hash engines (inner and outer loops) for each of the MD5 hash round logic and the SHA1 hash round logic supported by the IPSec protocol" [column 6 lines 18-25], which implies that pipelining is performed of both inner and outer loops.

Conclusion

9. **THIS ACTION IS MADE FINAL.** Applicant is reminded of the extension of time policy as set forth in 37 CFR 1.136(a).

A shortened statutory period for reply to this final action is set to expire THREE MONTHS from the mailing date of this action. In the event a first reply is filed within TWO MONTHS of the mailing date of this final action and the advisory action is not mailed until after the end of the THREE-MONTH shortened statutory period, then the shortened statutory period will expire on the date the advisory action is mailed, and any extension fee pursuant to 37 CFR 1.136(a) will be calculated from the mailing date of the advisory action. In no event, however, will the statutory period for reply expire later than SIX MONTHS from the mailing date of this final action.

Any inquiry concerning this communication or earlier communications from the examiner should be directed to Examiner Oscar Louie whose telephone number is 571-270-1684. The examiner can normally be reached Monday through Thursday from 7:30 AM to 4:00 PM.

If attempts to reach the examiner by telephone are unsuccessful, the examiner's supervisor, Nasser Moazzami, can be reached at 571-272-4195. The fax phone number for Formal or Official faxes to Technology Center 2100 is 571-273-8300.

Information regarding the status of an application may be obtained from the Patent Application Information Retrieval (PAIR) system. Status information for published applications may be obtained from either Private PAIR or Public PAIR. Status information for unpublished applications is available through Private PAIR only. For more information about the PAIR system, see http://pair-direct.uspto.gov. Should you have questions on access to the Private PAIR system, contact the Electronic Business Center (EBC) at 866-217-9197 (toll-free). If you would like assistance from a USPTO Customer Service Representative or access to the automated information system, call 800-786-9199 (IN USA OR CANADA) or 571-272-1000.

OAL
01/23/2007

Nasser Moazzami
Supervisory Patent Examiner

1/23/08